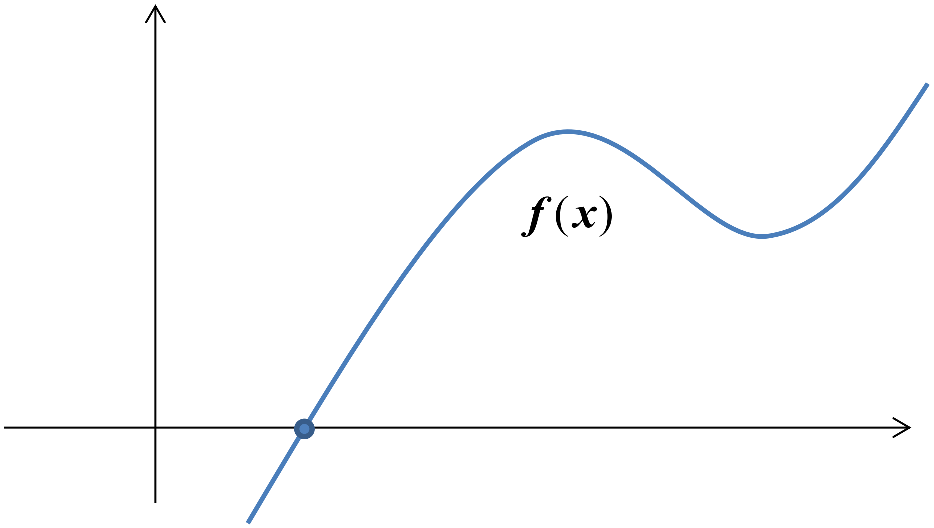
2º SEMESTRE: CÁLCULO NUMÉRICO CON FORTRAN.

TEMARIO:

1. Sistemas lineales. Directos e iterativos.
2. Cálculo de autovalores.
3. Derivación e integración numérica.
4. Solución de ecuaciones y sistemas no lineales.
5. Solución numérica de ecuaciones diferenciales.

RESOLUCIÓN NUMÉRICA DE ECUACIONES NO LINEALES



Motivación: Hallar los puntos x que verifican $f(x) = 0$

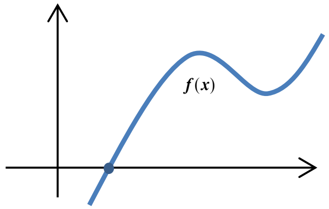
La no-linealidad de la ecuación hace que no sea posible en la mayoría de los casos hallar soluciones analíticas.

$$f(x) \equiv 2x^2 - 3x + 5 = 0$$

$$f(x) \equiv 4x^5 - x^3 - 2x^2 + 3 = 0$$

$$f(x) \equiv e^{3x} - \cos(x) = 0$$

RESOLUCIÓN NUMÉRICA DE ECUACIONES NO LINEALES



MÉTODOS NUMÉRICOS

Algoritmia

Los métodos numéricos en este caso se basan en criterios geométricos de aproximación a la solución mediante iteración.

Intervalos

Bisección

Regula Falsi

....

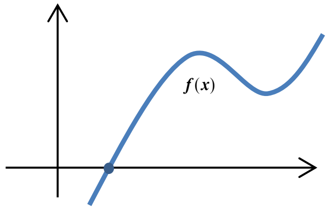
Valor Inicial

Newton

Secante

....

RESOLUCIÓN NUMÉRICA DE ECUACIONES NO LINEALES



MÉTODOS NUMÉRICOS

Algoritmia

Los métodos numéricos en este caso se basan en criterios geométricos de aproximación a la solución mediante iteración.

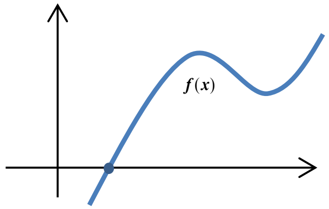
Intervalos

Bisección

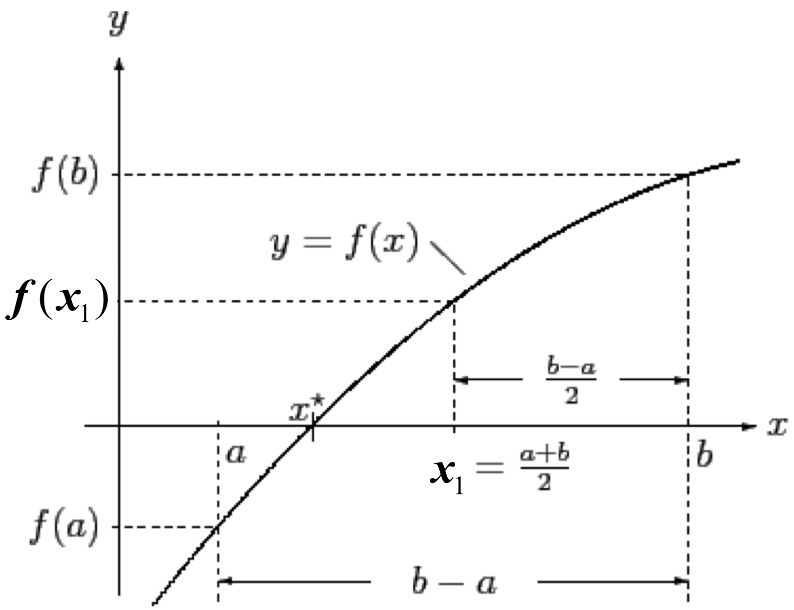
Regula Falsi

....

RESOLUCIÓN NUMÉRICA DE ECUACIONES NO LINEALES



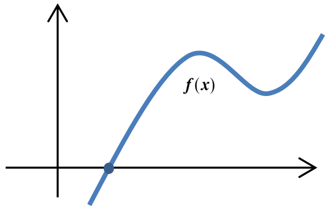
MÉTODO DE LA BISECCIÓN



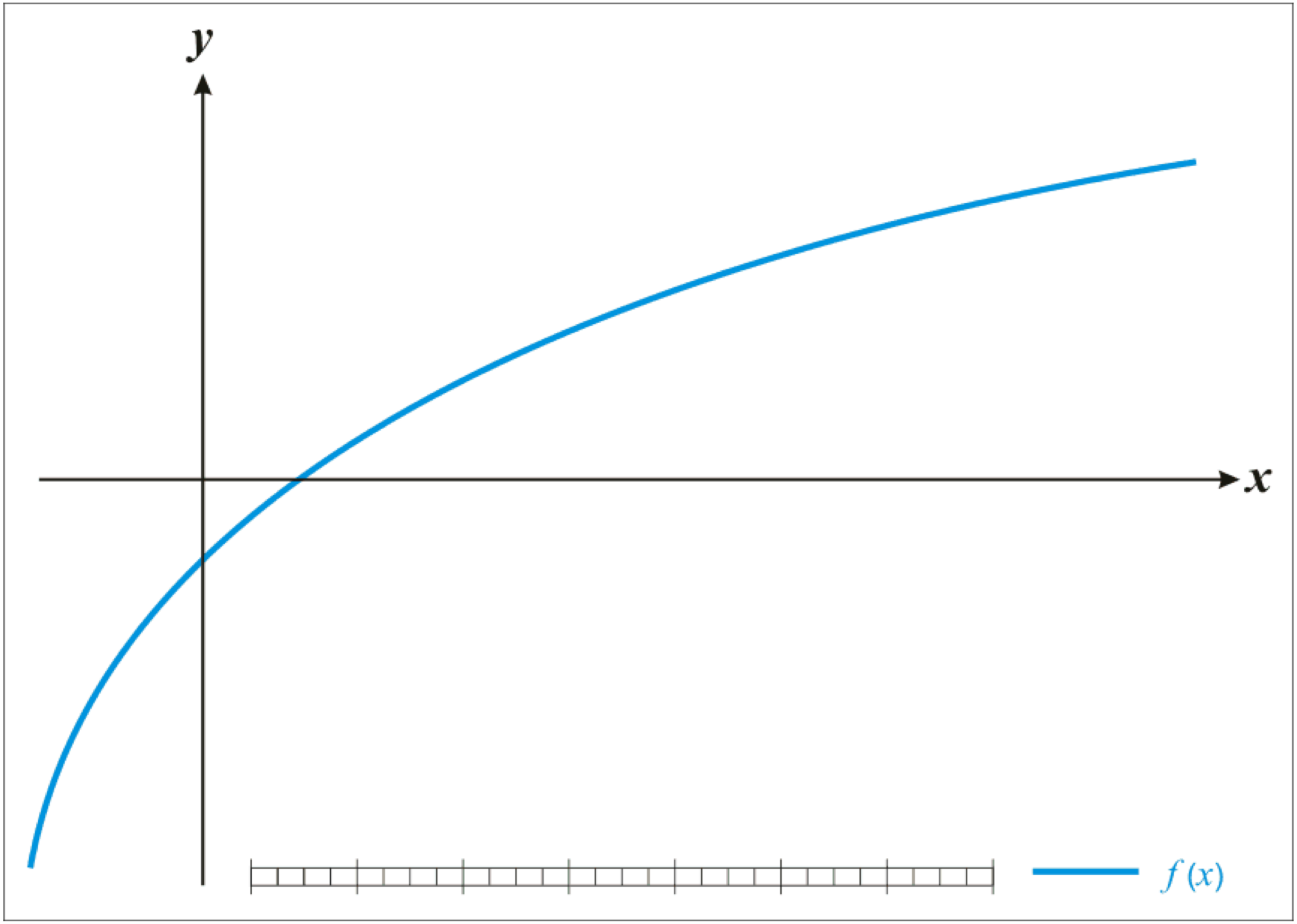
- Aplicación del Teorema de Bolzano: funciones continuas en intervalos cerrados, con signo diferente en los extremos del intervalo.

- Vamos dividiendo el intervalo por la mitad, manteniendo siempre la solución x en el intervalo.

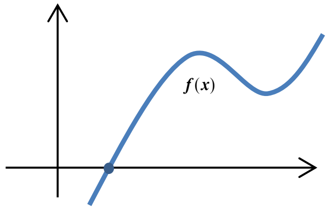
RESOLUCIÓN NUMÉRICA DE ECUACIONES NO LINEALES



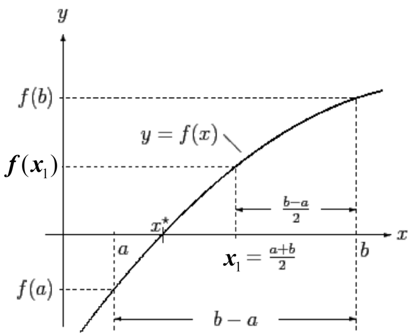
MÉTODO DE LA BISECCIÓN



RESOLUCIÓN NUMÉRICA DE ECUACIONES NO LINEALES



MÉTODO DE LA BISECCIÓN: Algoritmo



1. Establecer el intervalo (a,b) tal que $f(a)\cdot f(b)<0$ (signo distinto)

$$x_{i+1}=\frac{b+a}{2}$$

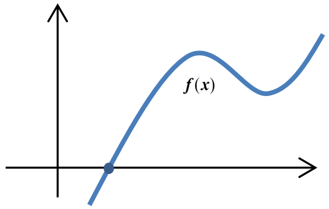
Si $\left|f(x_{i+1})\right|<Tol \rightarrow x^*=x_{i+1}$ → Fin

Si $f(x_{i+1})\cdot f(a)<0 \rightarrow \begin{cases} a=a \\ b=x_{i+1} \end{cases}$

Si $f(x_{i+1})\cdot f(b)<0 \rightarrow \begin{cases} a=x_{i+1} \\ b=b \end{cases}$

$i=i+1$

RESOLUCIÓN NUMÉRICA DE ECUACIONES NO LINEALES



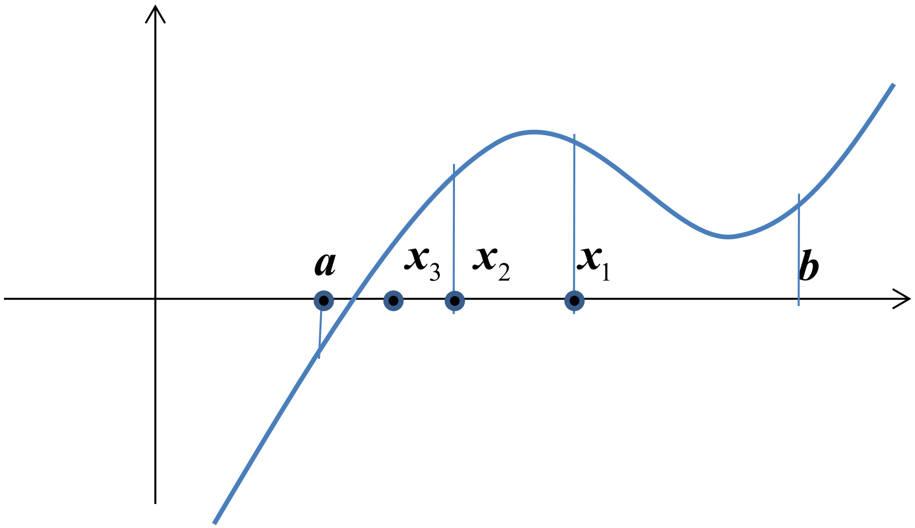
MÉTODO DE LA BISECCIÓN: Comentarios

- **Siempre converge.**

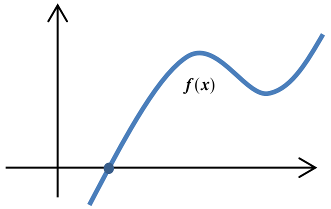
- **Muy sencillo de implementar, muy intuitivo geométricamente.**

- **Como poco, en cada iteración la incertidumbre se divide por dos.**

- La convergencia del método es lenta. Si el primer cálculo de la raiz es cercano a la verdadera, el método es extremadamente lento en converger.



RESOLUCIÓN NUMÉRICA DE ECUACIONES NO LINEALES



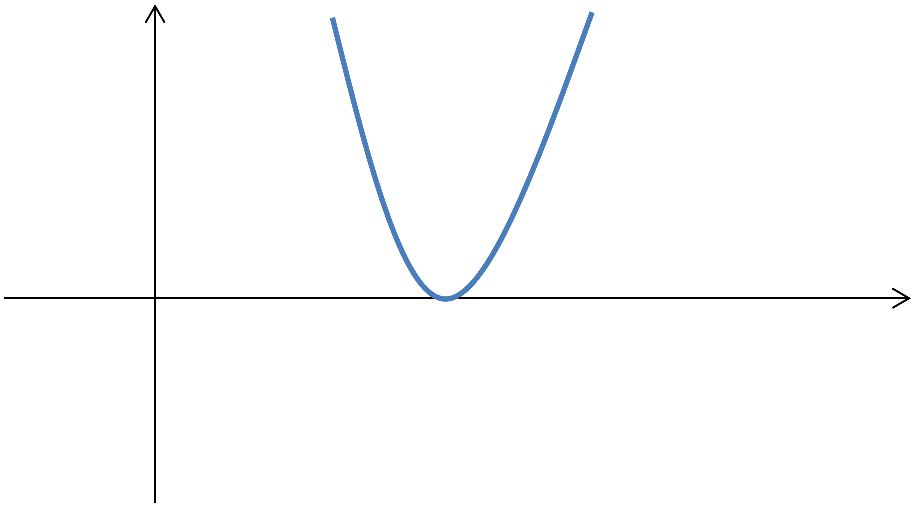
MÉTODO DE LA BISECCIÓN: Comentarios

- **Siempre converge.**

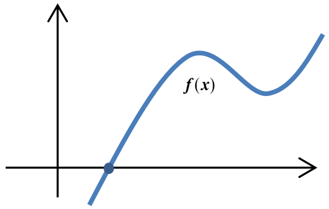
- **Muy sencillo de implementar, muy intuitivo geométricamente.**

- **Como poco, en cada iteración la incertidumbre se divide por dos.**

- Sólo es aplicable cuando la curva es secante al eje x (cuando la curva toma valores de signo contrario a un lado y otro de la raiz)

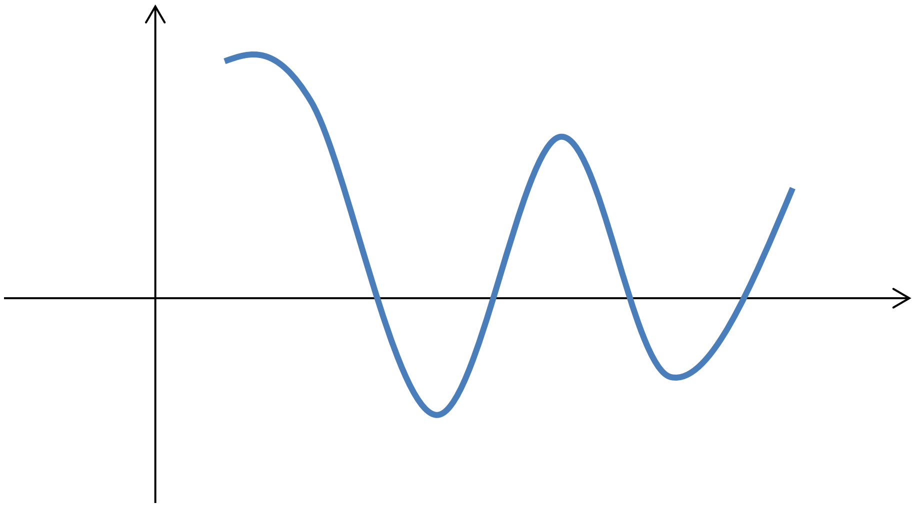


RESOLUCIÓN NUMÉRICA DE ECUACIONES NO LINEALES

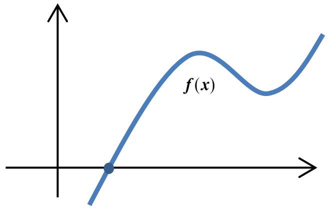


MÉTODO DE LA BISECCIÓN: Comentarios

- **Siempre converge.**
- **Muy sencillo de implementar, muy intuitivo geométricamente.**
- **Como poco, en cada iteración la incertidumbre se divide por dos.**
- En caso de multiples soluciones en el intervalo de definicion solo encuentra una.



RESOLUCIÓN NUMÉRICA DE ECUACIONES NO LINEALES



MÉTODOS NUMÉRICOS

Algoritmia

Los métodos numéricos en este caso se basan en criterios geométricos de aproximación a la solución mediante iteración.

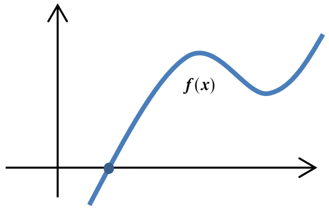
Valor Inicial

Newton

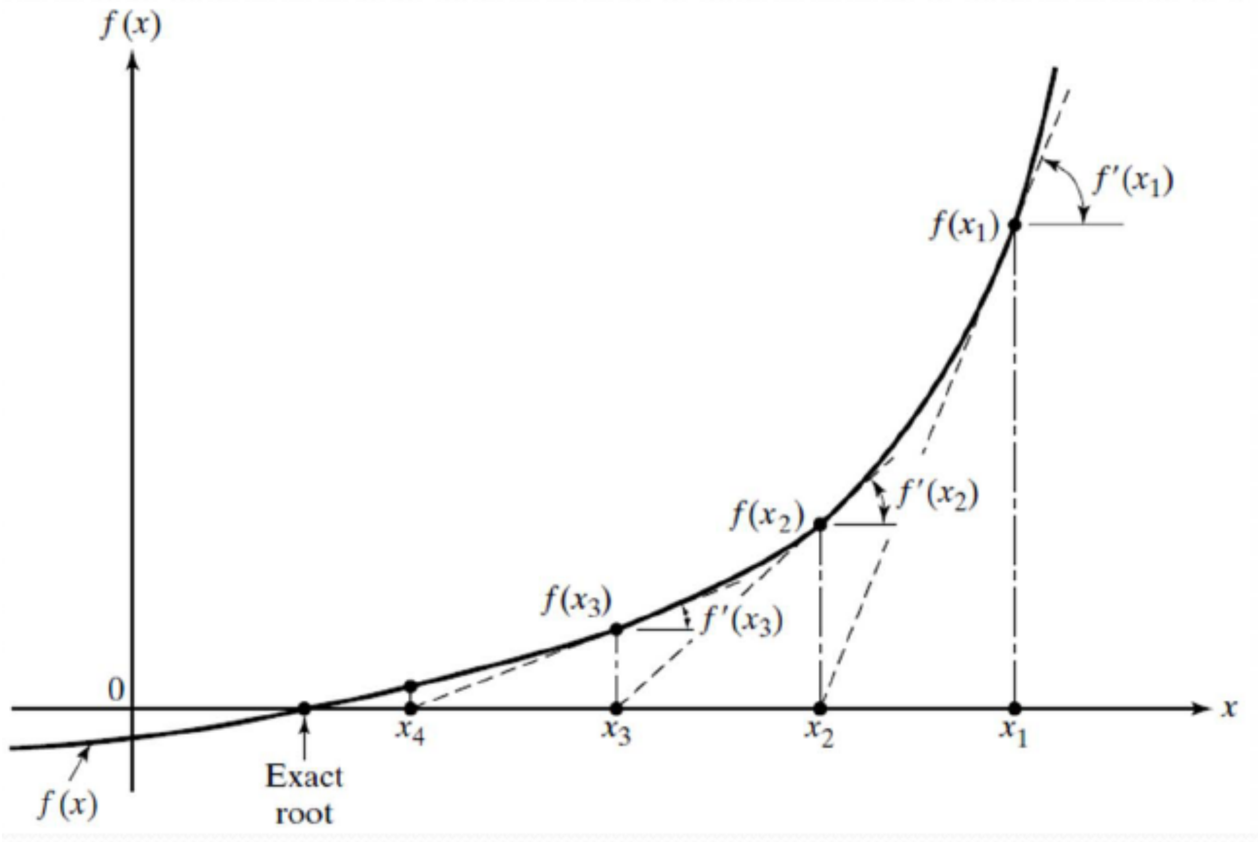
Secante

....

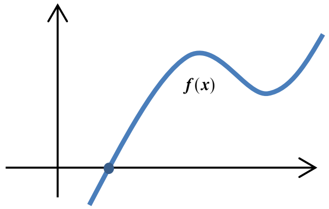
RESOLUCIÓN NUMÉRICA DE ECUACIONES NO LINEALES



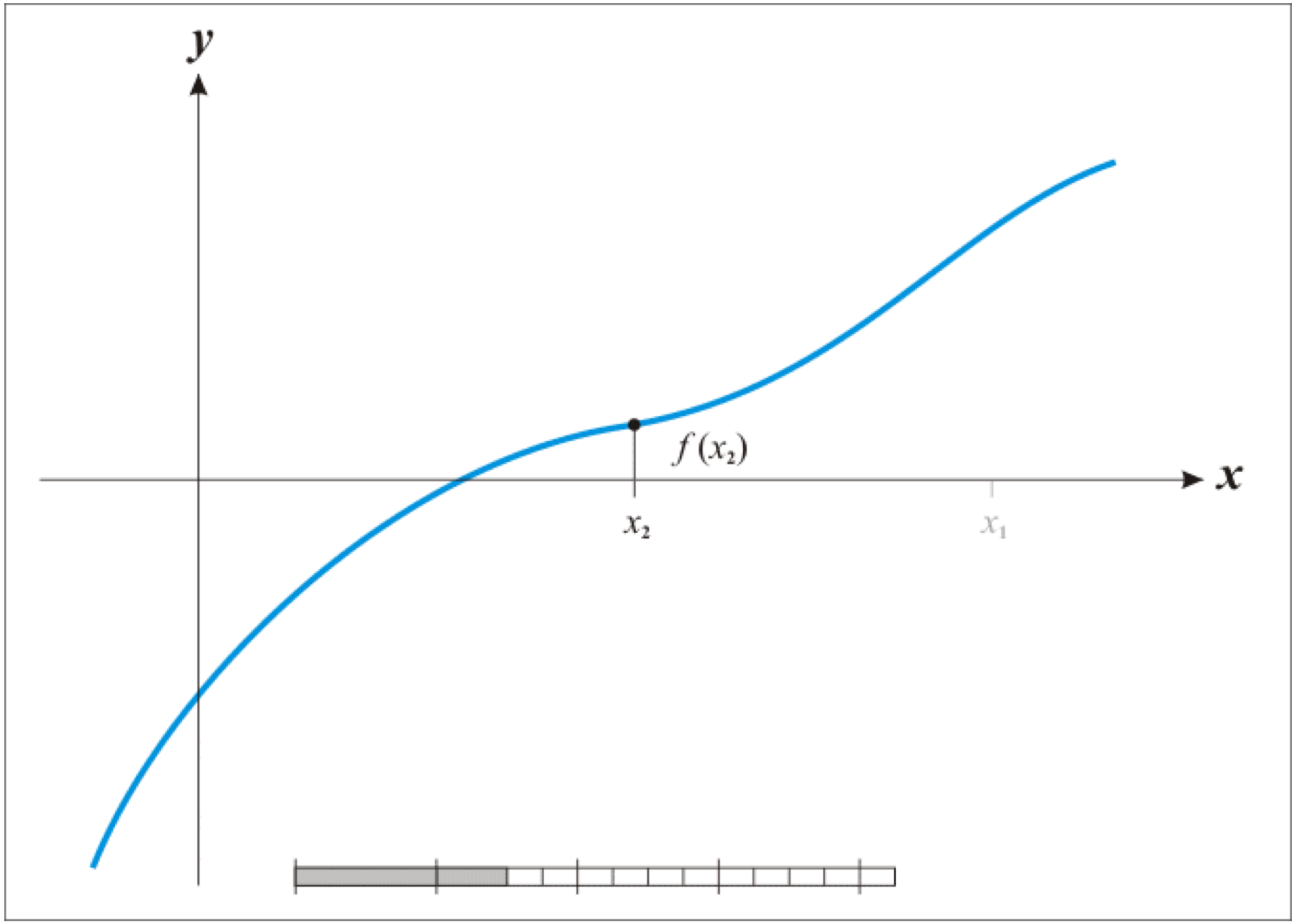
MÉTODO DE NEWTON (NEWTON-RAPHSON)



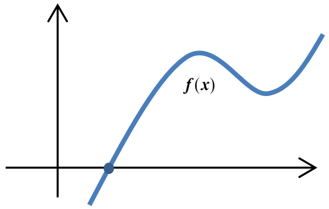
- El método de Newton-Raphson se basa en aproximar la función mediante la recta tangente. La función debe ser derivable.



MÉTODO DE NEWTON (NEWTON-RAPHSON)



RESOLUCIÓN NUMÉRICA DE ECUACIONES NO LINEALES



MÉTODO DE NEWTON: Algoritmo

Tomamos el desarrollo de Taylor de la función

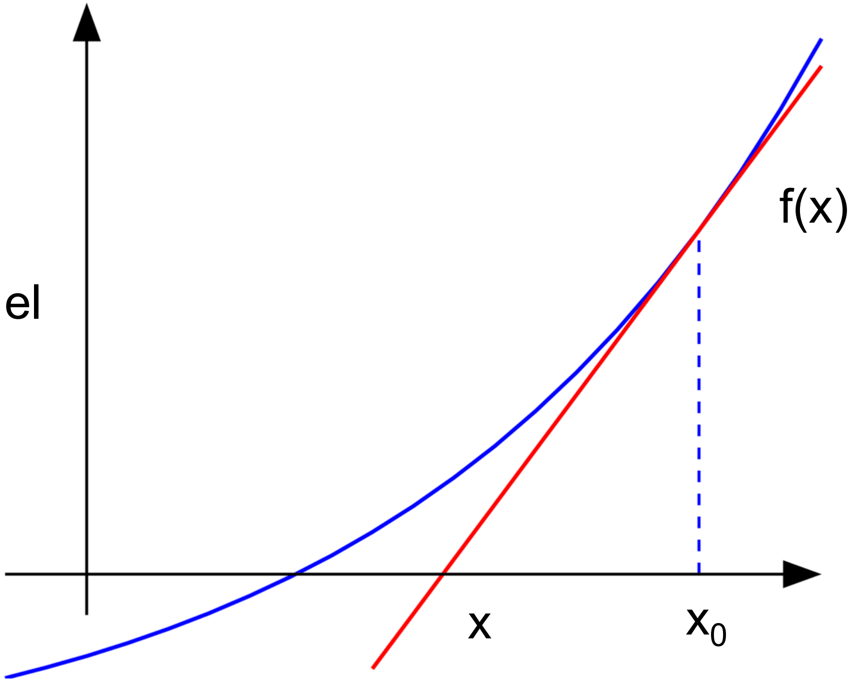
$$f(x) = f(x_0) + f'(x_0)(x - x_0) + \dots$$

Si tomamos solo los términos de primer orden, nos queda la ecuación de la recta tangente en el punto x_0. Para despejar el punto donde la tangente vale cero:

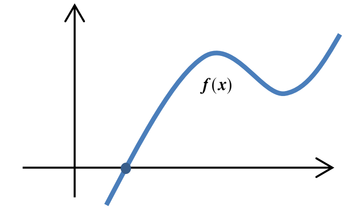
$$0 = f(x_0) + f'(x_0)(x - x_0)$$

Y despejando x

$$x = x_0 - \frac{f(x_0)}{f'(x_0)} \longrightarrow x_{i+1} = x_i - \frac{f(x_i)}{f'(x_i)}$$



RESOLUCIÓN NUMÉRICA DE ECUACIONES NO LINEALES



MÉTODO DE NEWTON: Algoritmo

Tomamos el desarrollo de Taylor de la función

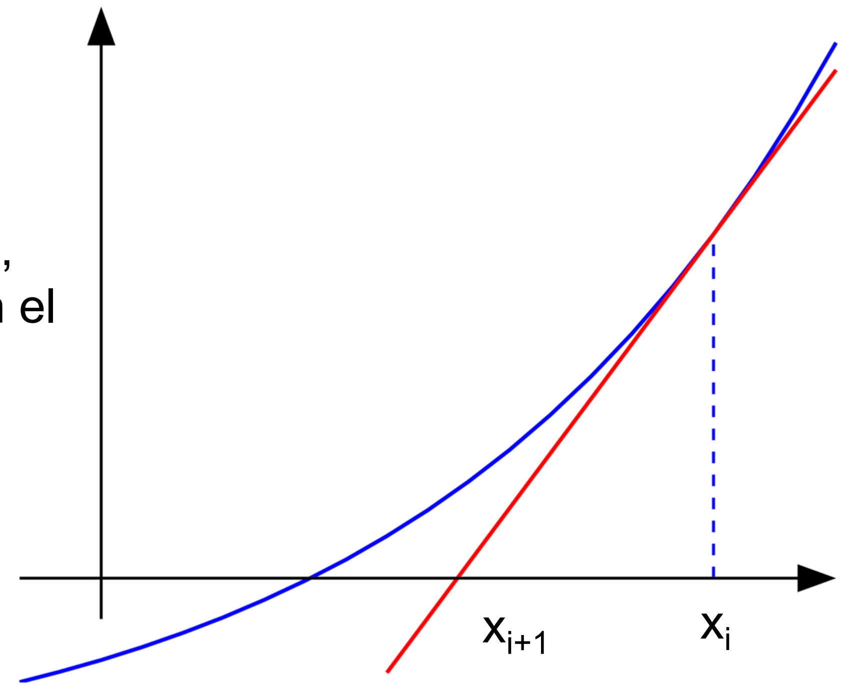
$$f(x) = f(x_0) + f'(x_0)(x - x_0) + \ldots$$

Si tomamos solo los términos de primer orden, nos queda la ecuación de la recta tangente en el punto x_0. Para despejar el punto donde la tangente vale cero:

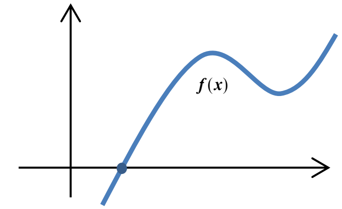
$$0 = f(x_0) + f'(x_0)(x - x_0)$$

Y despejando x

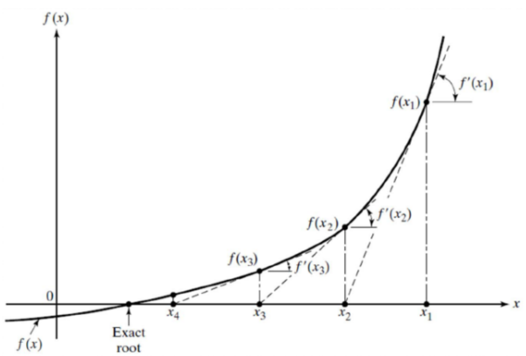
$$x = x_0 - \frac{f(x_0)}{f'(x_0)} \longrightarrow x_{i+1} = x_i - \frac{f(x_i)}{f'(x_i)}$$



RESOLUCIÓN NUMÉRICA DE ECUACIONES NO LINEALES



MÉTODO DE NEWTON: Algoritmo



Partimos de un punto inicial x_0

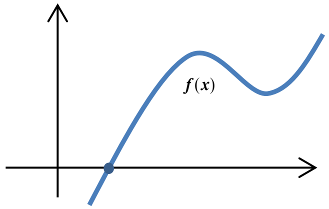
$$x_{i+1} = x_i - \frac{f(x_i)}{f'(x_i)}$$

$$\text{Si } \left| f(x_{i+1}) \right| < Tol \rightarrow x^* = x_{i+1}$$

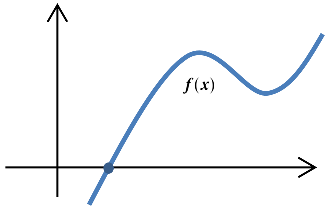
Fin

i=i+1

RESOLUCIÓN NUMÉRICA DE ECUACIONES NO LINEALES




```
subroutine newton(x,f,df,tol,max_iter)


real(8),intent(inout)      :: x
real(8),intent(in)         :: tol
integer,intent(in)         :: max_iter

interface

          function f(x)

                    real(8)     :: x
                    real(8)     :: f

          end function

          function df(x)

                    real(8)     :: x
                    real(8)     :: df

          end function

end interface
```



MÉTODO DE NEWTON: Comentarios

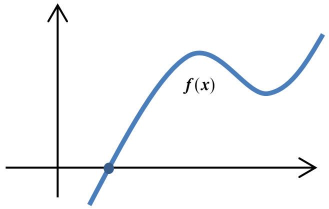
- **La convergencia del método es muy rápida (cuadrática) , en pocas iteraciones se obtiene una buena aproximación a la solución.**

- **Al contrario que el método de la bisección, este es aplicable para encontrar diversos tipos de raices, aunque la velocidad de convergencia puede variar en esos casos.**

- **La derivada puede conocerse de forma explícita (analítica) o aproximarse numéricamente (derivadas centradas, progresivas, regresivas...)**

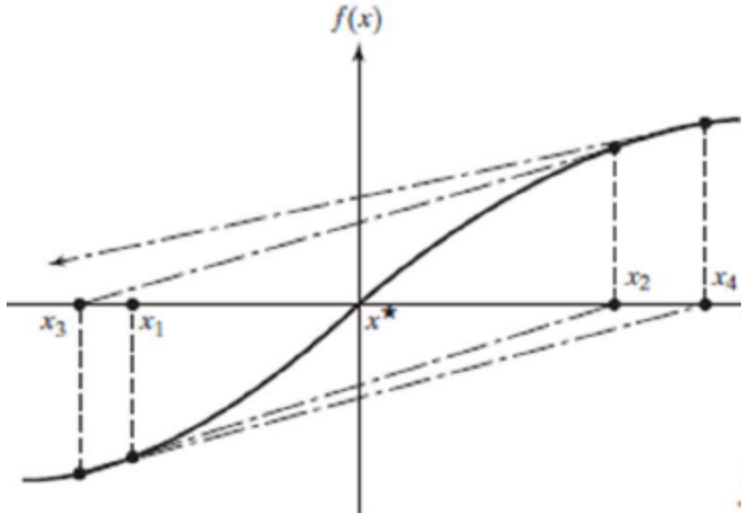
- **Coincide con *Regula Falsi* si se hace:**

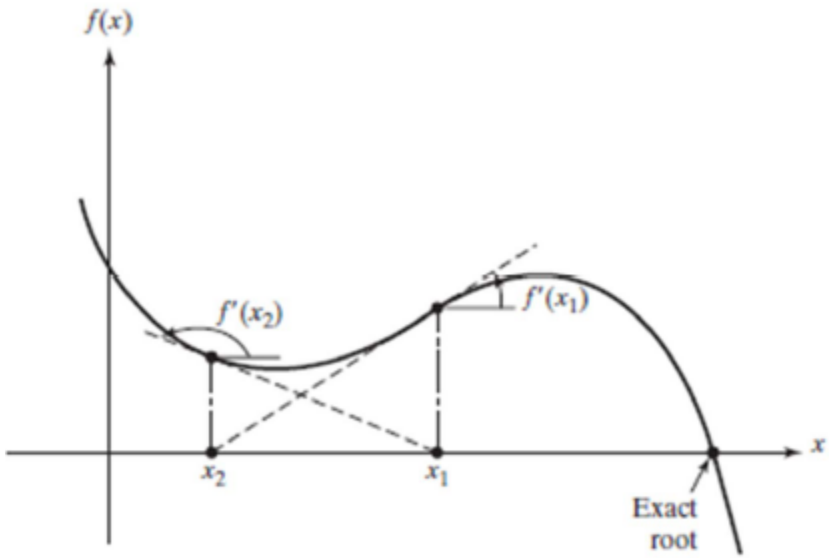
$$f'(x_{i+1}) = \frac{f(x_{i+1}) - f(x_i)}{x_{i+1} - x_i}$$

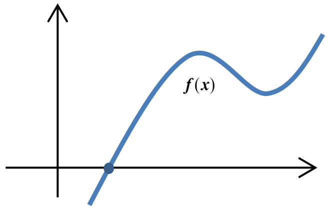


MÉTODO DE NEWTON: Comentarios

- NO siempre converge (convergencia local): la condición inicial ha de estar relativamente cerca de la raiz verdadera

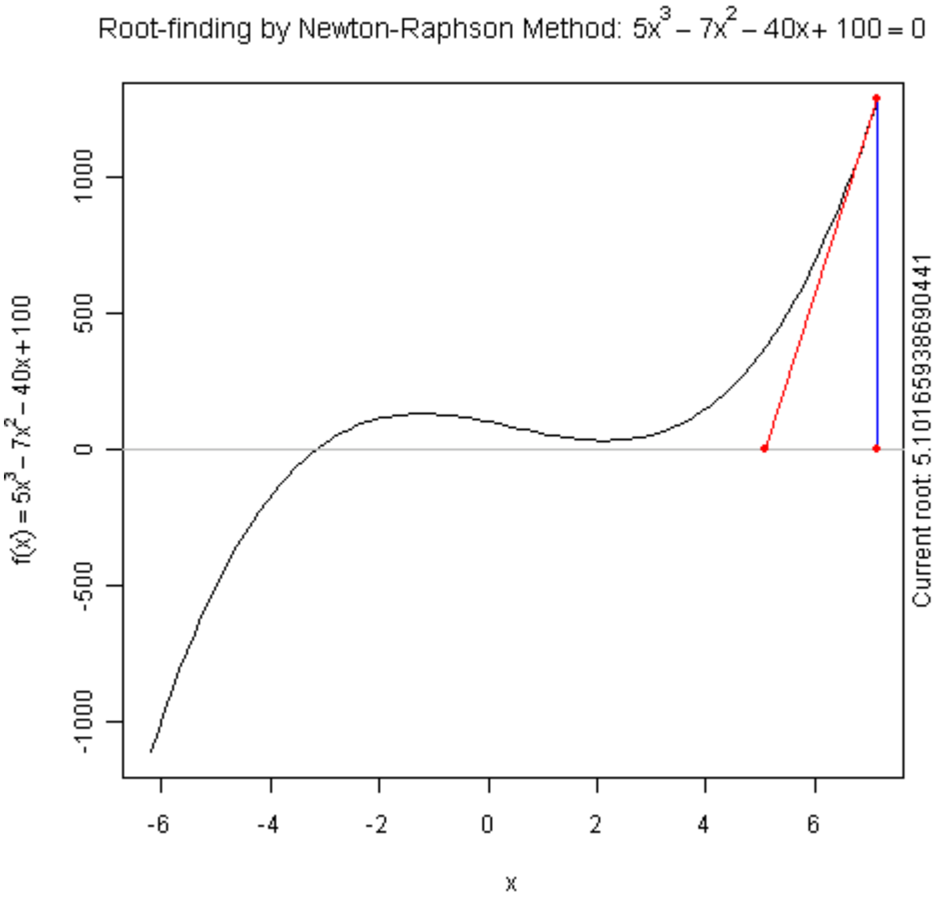


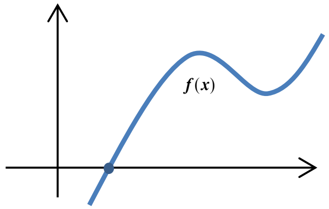




MÉTODO DE NEWTON: Comentarios

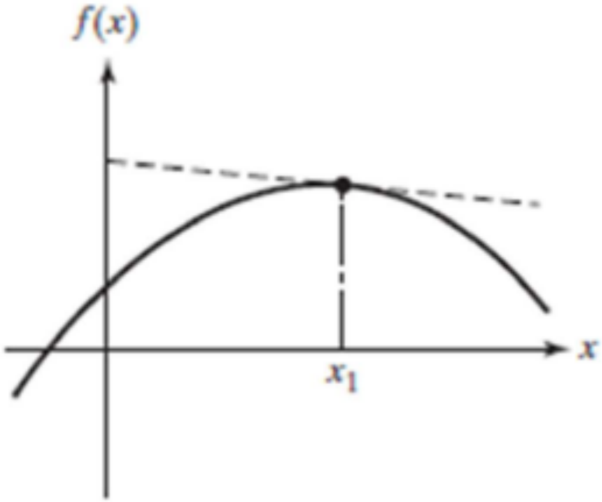
- NO siempre converge (convergencia local): la condición inicial ha de estar relativamente cerca de la raiz verdadera

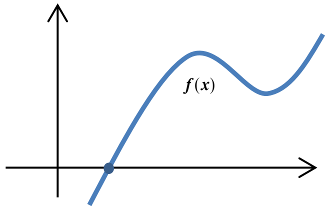




MÉTODO DE NEWTON: Comentarios

- La derivada de la función ha de conocerse explícitamente y no puede anularse en un entorno de la raiz



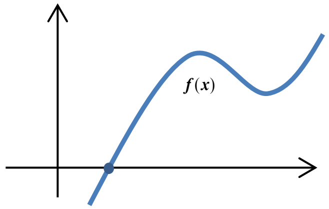


MÉTODOS NUMÉRICOS: Criterio de parada

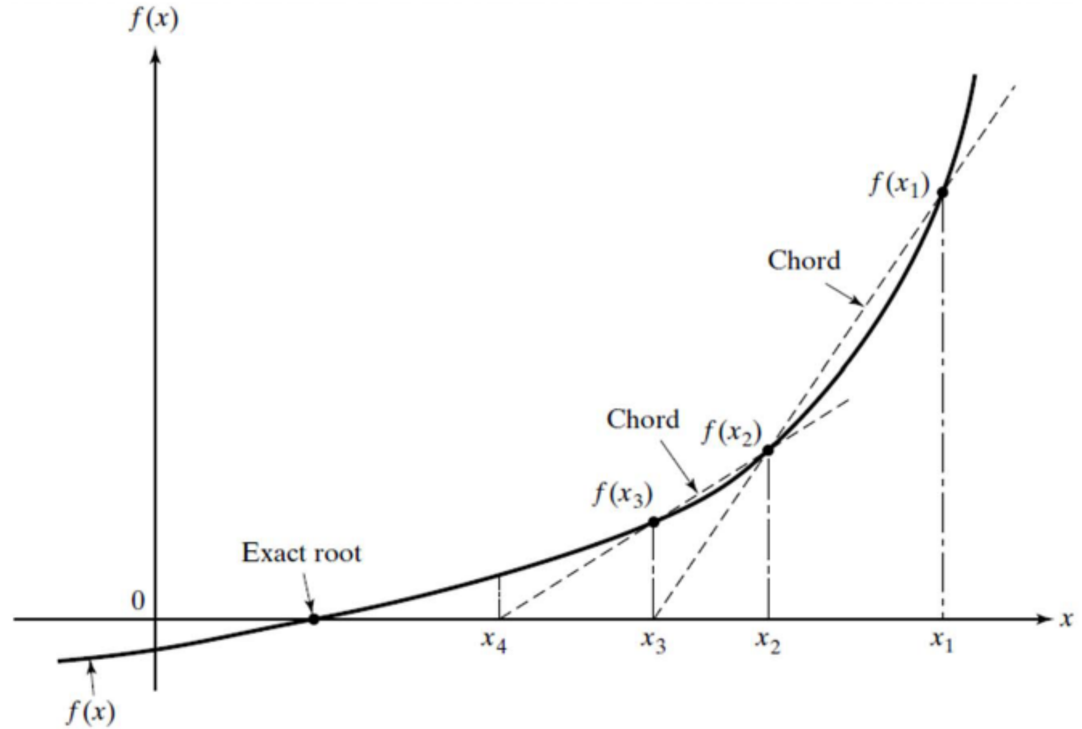
- Número máximo de iteraciones.
- Valor suficientemente próximo: Si $\left|f(x_{i+1})\right| < Tol \rightarrow x^{*} = x_{i+1}$
- Error relativo:

$$\varepsilon_{rel} = \left|\frac{x - x_{i+1}}{x}\right| < Tol \rightarrow x^{*} = x_{i+1} \qquad \varepsilon_{rel} = \left|\frac{x_{i+1} - x_{i}}{x_{i+1}}\right| < Tol \rightarrow x^{*} = x_{i+1}$$

RESOLUCIÓN NUMÉRICA DE ECUACIONES NO LINEALES



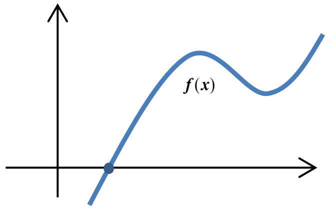
Complemento 1: MÉTODO DE REGULA FALSI Y MÉTODO DE LA SECANTE



Explicar en clase, con diapositivas, el método de la secante y el método de regula falsi y su relación con otros métodos vistos en clase.
Programar y comparar con el resto de métodos vistos en clase.

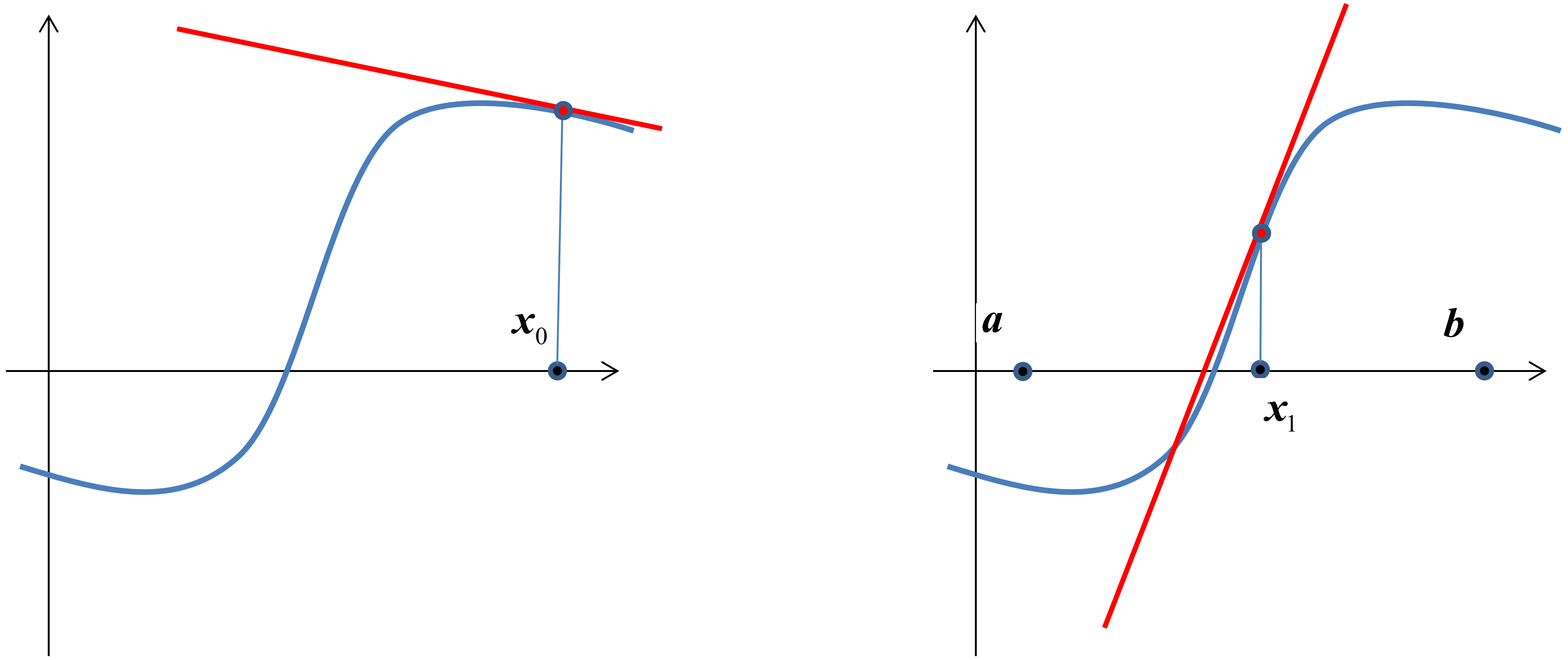
https://es.wikipedia.org/wiki/Método_de_la_regla_falsa

https://es.wikipedia.org/wiki/Método_de_la_secante

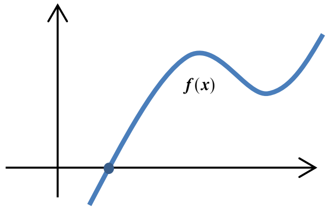


Complemento 2: MÉTODO DE REGULA FALSI Y MÉTODO DE LA SECANTE

- **Método optimizado: usar bisección para las primeras iteraciones (convergencia global pero lenta), y usar Newton una vez estemos cerca de la raiz (convergencia local pero rápida).**



RESOLUCIÓN NUMÉRICA DE ECUACIONES NO LINEALES

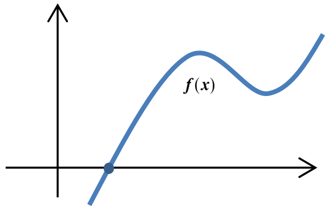


Práctica para todos los grupos

1- Programad los distintos métodos vistos en clase.

- Bisección
- Newton
- Newton con derivada numérica

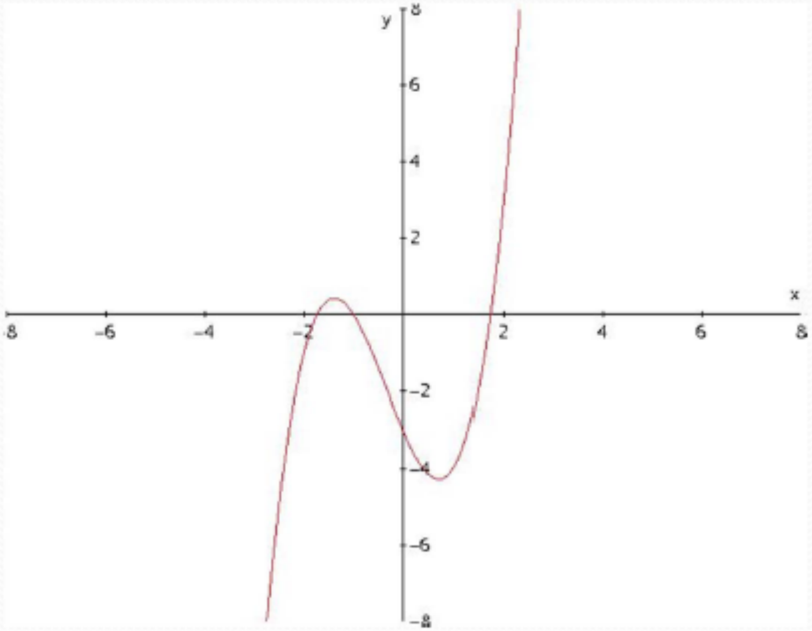
2.- Encontrar los ceros de la función $g(x) = x^3 + x^2 - 3x - 3$



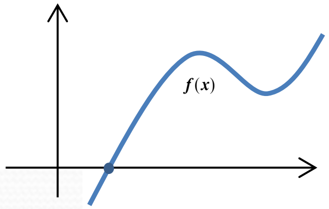
- Vamos a programar algoritmos para:

$$f(x) = x^3 + x^2 - 3x - 3 \quad \text{en } [0,10]$$

- Utilizar $\varepsilon<10^{-6}$ (error relativo de x)



RESOLUCIÓN NUMÉRICA DE ECUACIONES NO LINEALES



- Vamos a programar algoritmos para:

$$f(x) = x^3 + x^2 - 3x - 3$$

Metodo de la bisección [a,b]=[0,5]

Iter	x	f(x)	error relativo
1	2.5000000000	11.3750000000	1.0000000000
2	1.2500000000	-3.2343750000	1.0000000000
3	1.8750000000	1.4824218750	0.3333333333
4	1.5625000000	-1.4313964844	0.2000000000
5	1.7187500000	-0.1247863770	0.0909090909
6	1.7968750000	0.6398124695	0.0434782609
7	1.7578125000	0.2479405403	0.0222222222
8	1.7382812500	0.0592063069	0.0112359551
9	1.7285156250	-0.0333799347	0.0056497175
10	1.7333984375	0.0127653619	0.0028169014
11	1.7309570312	-0.0103441988	0.0014104372
12	1.7321777344	0.0012013480	0.0007047216
13	1.7315673828	-0.0045737331	0.0003524850
14	1.7318725586	-0.0016867696	0.0001762115
15	1.7320251465	-0.0002428550	0.0000880980
16	1.7321014404	0.0004792104	0.0000440470
17	1.7320632935	0.0001181687	0.0000220240
18	1.7320442200	-0.0000623454	0.0000110121
19	1.7320537567	0.0000279111	0.0000055060
20	1.7320489883	-0.0000172173	0.0000027530
21	1.7320513725	0.0000053468	0.0000013765
22	1.7320501804	-0.0000059353	0.0000006883
23	1.7320507765	-0.0000002942	0.0000003441

Solucion encontrada 1.73205077648163

Metodo de Newton X0=10

Iter	x	f(x)	error relativo
1	6.6340694006	313.0798797665	0.5073704232
2	4.4339415908	90.5285867685	0.4962013515
3	3.0379166063	25.1519293916	0.4595336757
4	2.2203039049	6.2143796248	0.3682435993
5	1.8374058935	1.0670360452	0.2083905427
6	1.7386336328	0.0625693137	0.0568102784
7	1.7320789950	0.0002667739	0.0037842603
8	1.7320508081	0.0000000049	0.0000162737
9	1.7320508076	0.0000000000	0.0000000003

Solucion encontrada 1.73205080756888